ACM PORTAL
USPTO
Subscribe (Full Service) Register (Limited Service, Free) Login
Search: The ACM Digital Library The Guide
"inout socket" and "output socket" and ALU and "aliging operai
SEARCH
THE ACM DIGITAL LIBRARY
Feedback Report a problem Satisfaction survey

Terms used **inout socket** and **output socket** and **ALU** and **aliging operands** and **lower significant** and **register** and **mask** and **memory**

Found **3,532** of **176,279**

Sort results by: relevance
Display results: expanded form
Save results to a Binder
Search Tips
Open results in a new window

Try an Advanced Search
Try this search in The ACM Guide

Results 1 - 20 of 200 Result page: **1** 2 3 4 5 6 7 8 9 10 next
Best 200 shown Relevance scale

1 RISCY patents
David A. Patterson
September 1988 **ACM SIGARCH Computer Architecture News**, Volume 16 Issue 4
Publisher: ACM Press
Full text available: pdf(1.83 MB) Additional Information: full citation, index terms

2 Architectural support for reduced register saving/restoring in single-window register files
Miquel Huguet, Tomás Lang
February 1991 **ACM Transactions on Computer Systems (TOCS)**, Volume 9 Issue 1
Publisher: ACM Press
Full text available: pdf(2.28 MB) Additional Information: full citation, abstract, references, citings, index terms, review

The use of registers in a processor reduces the data and instruction memory traffic. Since this reduction is a significant factor in the improvement of the program execution time, recent VLSI processors have a large number of registers which can be used efficiently because of the advances in compiler technology. However, since registers have to be saved/restored across function calls, the corresponding register saving and restoring (RSR) memory traffic can almost eliminate the overall reduc ...

3 Our machine, a microcoded LSI processor
Dave Johannsen
November 1978 **ACM SIGMICRO Newsletter , Proceedings of the 11th annual workshop on Microprogramming MICRO 11**, Volume 9 Issue 4
Publisher: IEEE Press, ACM Press
Full text available: pdf(1.38 MB) Additional Information: full citation, abstract, citings, index terms

Current LSI technology allows the systems designer to construct complex data processing structures containing tens of thousands of transistors on single silicon chips. Constraints imposed by the technology influence design tradeoffs and result in computer architectures dramatically different from the more classical computer designs. At CalTech we are exploring the possibilities offered by nMOS technology, with the "Our Machine" (OM) project being one of the current research proj ...

4 A study on the number of memory ports in multiple instruction issue machines
Soo-Mook Moon, Kemal Ebcioğlu
December 1993 **Proceedings of the 26th annual international symposium on Microarchitecture**
Publisher: IEEE Computer Society Press
Full text available: pdf(1.28 MB) Additional Information: full citation, references, citings

Keywords: ILP, memory disambiguation, memory ports, speculative loads, static scheduling

5 Reduced register saving/restoring in single-window register files
Tomás Lang, Miquel Huguet
June 1986 **ACM SIGARCH Computer Architecture News**, Volume 14 Issue 3
Publisher: ACM Press
Full text available: pdf(751.89 KB) Additional Information: full citation, citings, index terms

6 Microprogrammable microprocessor survey
Phillip M. Adams
June 1978 **ACM SIGMICRO Newsletter**, Volume 9 Issue 2
Publisher: ACM Press
Full text available: pdf(1.24 MB) Additional Information: full citation, abstract

The Motorola M10800 LSI processor family consists of a sequencer, referred to as a Microprogram Control Function (MCF) - MC10801, and a processing element, referred to as a 4-bit ALU Slice - MC10800 (Not to be confused with the processor family number M10800). Undoubtedly, the most interesting feature of the M10800 processor family is the ECL technology used to produce it. The M10800 processor family is completely MECL 10,000 compatible and exhibits the ultra-high speed performance of ECL logic.

7 Gilgamesh: a multithreaded processor-in-memory architecture for petaflops computing
Thomas L. Sterling, Hans P. Zima
November 2002 **Proceedings of the 2002 ACM/IEEE conference on Supercomputing**
Publisher: IEEE Computer Society Press
Full text available: pdf(322.86 KB) Additional Information: full citation, abstract, references, citings, index terms

Processor-in-Memory (PIM) architectures avoid the von Neumann bottleneck in conventional machines by integrating high-density DRAM and CMOS logic on the same chip. Parallel systems based on this new technology are expected to provide higher scalability, adaptability, robustness, fault tolerance and lower power consumption than current MPPs or commodity clusters. In this paper we describe the design of *Gilgamesh*, a PIM-based massively parallel architecture, and elements of its execution mo ...

Keywords: Petaflops computing, Processor-In-Memory, data parallel processing, irregular applications, parallel architectures

8 Pioneer: verifying code integrity and enforcing untampered code execution on legacy systems

Arvind Seshadri, Mark Luk, Elaine Shi, Adrian Perrig, Leendert van Doorn, Pradeep Khosla
October 2005 **ACM SIGOPS Operating Systems Review , Proceedings of the twentieth ACM symposium on Operating systems principles SOSP '05**, Volume 39 Issue 5
Publisher: ACM Press
Full text available: pdf(264.30 KB) Additional Information: full citation, abstract, references, index terms

We propose a primitive, called Pioneer, as a first step towards verifiable code execution on untrusted legacy hosts. Pioneer does not require any hardware support such as secure co-processors or CPU-architecture extensions. We implement Pioneer on an Intel Pentium IV Xeon processor. Pioneer can be used as a basic building block to build security systems. We demonstrate this by building a kernel rootkit detector.

Keywords: dynamic root of trust, rootkit detection, self-check-summing code, software-based code attestation, verifiable code execution

9 GPGPU: general purpose computation on graphics hardware

David Luebke, Mark Harris, Jens Krüger, Tim Purcell, Naga Govindaraju, Ian Buck, Cliff Woolley, Aaron Lefohn
August 2004 **Proceedings of the conference on SIGGRAPH 2004 course notes GRAPH '04**
Publisher: ACM Press
Full text available: pdf(63.03 MB) Additional Information: full citation, abstract

The graphics processor (GPU) on today's commodity video cards has evolved into an extremely powerful and flexible processor. The latest graphics architectures provide tremendous memory bandwidth and computational horsepower, with fully programmable vertex and pixel processing units that support vector operations up to full IEEE floating point precision. High level languages have emerged for graphics hardware, making this computational power accessible. Architecturally, GPUs are highly parallel s ...

10 Towards an efficient, machine-independent language for microprogramming

David A. Patterson, Karl Lew, Richard Tuck
November 1979 **ACM SIGMICRO Newsletter , Proceedings of the 12th annual workshop on Microprogramming MICRO 12**, Volume 10 Issue 4
Publisher: IEEE Press, ACM Press
Full text available: pdf(913.17 KB) Additional Information: full citation, abstract, references, citings, index terms

A machine independent low level language YALLL is presented. This language produces microcode for two very different machines: Hewlett Packard HP 300 and Digital Equipment Corporation VAX 11/780. The efficiency of this language is tested by comparing two examples on both machines to microassembly coded versions. To our best knowledge, this is the first time programs have been compiled and executed on two different microarchitectures. These examples also let us compare the efficiency of the ...

11 A frame buffer system with enhanced functionality

F. C. Crow, M. W. Howard
August 1981 **ACM SIGGRAPH Computer Graphics , Proceedings of the 8th annual conference on Computer graphics and interactive techniques SIGGRAPH '81**, Volume 15 Issue 3
Publisher: ACM Press
Full text available: pdf(561.14 KB) Additional Information: full citation, abstract, references, index terms

A video-resolution frame buffer system with 32 bits per pixel is described. The system includes, in addition to standard features for limited zoom and pan, an arithmetic unit at the update port which allows local computation of many frequently-used pixel-level

functions combining stored pixel values with incoming pixel values. In addition to the standard arithmetic and logical functions there are functions for sum to maximum pixel value and difference to minimum pixel value. Comparisons bet ...

12 The Postroom Computer

Hugh Osborne
December 2001 **Journal on Educational Resources in Computing (JERIC)**, Volume 1 Issue 4
Publisher: ACM Press
Full text available: pdf(242.80 KB) Additional Information: full citation, abstract, references, index terms

The Postroom Computer is a computer architecture simulator based on the *Little Man Computer* developed in 1965 by Stuart Madnick and John Donovan. It provides a family of architectures suitable for use in teaching introductory computer architectures. It is designed to introduce aspects of computer architecture and low-level programming in an incremental way. The extensions are designed to provide a range of computing models within the Little Man Computer paradigm. As they are introduced th ...

Keywords: Computer architecture simulator, education

13 Transient-fault recovery for chip multiprocessors

Mohamed Gomaa, Chad Scarbrough, T. N. Vijaykumar, Irith Pomeranz
May 2003 **ACM SIGARCH Computer Architecture News , Proceedings of the 30th annual international symposium on Computer architecture ISCA '03**, Volume 31 Issue 2
Publisher: ACM Press
Full text available: pdf(370.75 KB) Additional Information: full citation, abstract, references, citings

To address the increasing susceptibility of commodity chip multiprocessors (CMPs) to transient faults, we propose Chiplevel Redundantly Threaded multiprocessor with Recovery (CRTR). CRTR extends the previously-proposed CRT for transient-fault detection in CMPs, and the previously-proposed SRTR for transient-fault recovery in SMT. All these schemes achieve fault tolerance by executing and comparing two copies, called leading and trailing threads, of a given application. Previous recovery schemes ...

14 Session 4C: Computer architecture: A new simulator workbench for comparing SIMD processing element architectures

Todd C. Marek
April 1992 **Proceedings of the 30th annual Southeast regional conference**
Publisher: ACM Press
Full text available: pdf(608.48 KB) Additional Information: full citation, abstract, references

The impact of machine structure on system performance is a critical consideration in designing highly integrated SIMD architectures. This issue is affected by PE granularity, PE complexity, and interconnection structure. Detailed analysis of the issues related to this structure/performance relationship is relevant in developing new massively parallel architectures. To meet this need, a simulator workbench which can be used to quantitatively evaluate the performance of a wide range of machine str ...

15 Memory interfacing and instruction specification for reconfigurable processors

Jeffrey A. Jacob, Paul Chow
February 1999 **Proceedings of the 1999 ACM/SIGDA seventh international symposium on Field programmable gate arrays**
Publisher: ACM Press
Full text available: pdf(1.77 MB) Additional Information: full citation, references, citings, index terms

Keywords: FPGA, memory coherence, memory interfacing, reconfigurable computer, reconfigurable processor

16 A processor for a high-performance personal computer

Butler W. Lampson, Kenneth A. Pier
May 1980 **Proceedings of the 7th annual symposium on Computer Architecture**
Publisher: ACM Press

Full text available: pdf(1.24 MB) Additional Information: full citation, abstract, references, citings, index terms

This paper describes the design goals, micro- architecture, and implementation of the microprogrammed processor for a compact high performance personal computer. This computer supports a range of high level language environments and high bandwidth I/O devices. Besides the processor, it has a cache, a memory map, main storage, and an instruction fetch unit; these are described in other papers. The processor can be shared among 16 microcoded tasks, performing microcode context switches ...

17 TRIPS: A polymorphous architecture for exploiting ILP, TLP, and DLP

Karthikeyan Sankaralingam, Ramadass Nagarajan, Haiming Liu, Changkyu Kim, Jaehyuk Huh, Nitya Ranganathan, Doug Burger, Stephen W. Keckler, Robert G. McDonald, Charles R. Moore
March 2004 **ACM Transactions on Architecture and Code Optimization (TACO)**, Volume 1 Issue 1
Publisher: ACM Press
Full text available: pdf(832.30 KB) Additional Information: full citation, abstract, references, index terms

This paper describes the *polymorphous* TRIPS architecture that can be configured for different granularities and types of parallelism. The TRIPS architecture is the first in a class of post-RISC, dataflow-like instruction sets called explicit data-graph execution (EDGE). This EDGE ISA is coupled with hardware mechanisms that enable the processing cores and the on-chip memory system to be configured and combined in different modes for instruction, data, or thread-level parallelism. To adapt ...

Keywords: Computer architecture, configurable computing, scalable and high-performance computing

18 Architecture: The architecture of the DIVA processing-in-memory chip

Jeff Draper, Jacqueline Chame, Mary Hall, Craig Steele, Tim Barrett, Jeff LaCoss, John Granacki, Jaewook Shin, Chun Chen, Chang Woo Kang, Ihn Kim, Gokhan Daglikoca
June 2002 **Proceedings of the 16th international conference on Supercomputing**
Publisher: ACM Press
Full text available: pdf(295.98 KB) Additional Information: full citation, abstract, citings, index terms

The DIVA (Data IntensiVe Architecture) system incorporates a collection of Processing-In-Memory (PIM) chips as smart-memory co-processors to a conventional microprocessor. We have recently fabricated prototype DIVA PIMs. These chips represent the first smart-memory devices designed to support virtual addressing and capable of executing multiple threads of control. In this paper, we describe the prototype PIM architecture. We emphasize three unique features of DIVA PIMs, namely, the memory interf ...

Keywords: architecture, memory bandwidth, processing-in-memory

19 A reconfigurable arithmetic array for multimedia applications

Alan Marshall, Tony Stansfield, Igor Kostarnov, Jean Vuillemin, Brad Hutchings
February 1999 **Proceedings of the 1999 ACM/SIGDA seventh international symposium on Field programmable gate arrays**
Publisher: ACM Press
Full text available: pdf(2.02 MB) Additional Information: full citation, references, citings, index terms

Keywords: 4-bit ALU, FPGA, multimedia, reconfigurable computing

20 Architectural tradeoffs in the design of MIPS-X

P. Chow, M. Horowitz
June 1987 **Proceedings of the 14th annual international symposium on Computer architecture**
Publisher: ACM Press
Full text available: pdf(943.92 KB) Additional Information: full citation, abstract, references, citings, index terms

The design of a RISC processor requires a careful analysis of the tradeoffs that can be made between hardware complexity and software. As new generations of processors are built to take advantage of more advanced technologies, new and different tradeoffs must be considered. We examine the design of a second generation VLSI RISC processor, MIPS-X. MIPS-X is the successor to the MIPS project at Stanford University and like MIPS, it is a single-chip 32-bit VLSI processor that uses a ...

The ACM Portal is published by the Association for Computing Machinery. Copyright © 2006 ACM, Inc.
Terms of Usage Privacy Policy Code of Ethics Contact Us

Useful downloads: Adobe Acrobat QuickTime Windows Media Player Real Player

	Type	L #	Hits	Search Text	DBs	Time Stamp
1	BRS	L1	70944	data adj packet	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	2006/05/05 09:48
2	BRS	L2	54656	1 and network	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	2006/05/05 09:38
3	BRS	L3	16060	2 and decode$	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	2006/05/05 09:39
4	BRS	L4	34268	network near3 processor	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	2006/05/05 09:39
5	BRS	L5	1885	3 and 4	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	2006/05/05 09:39
6	BRS	L6	23359198	@ad<"20021212"	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	2006/05/05 09:40

7	BRS	L7	1381	5 and 6	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	2006/05/05 09:47

	Type	L #	Hits	Search Text	DBs	Time Stamp
8	BRS	L8	442	7 and align$	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	2006/05/05 09:43
9	BRS	L9	1	7 and align$ near3 operands	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	2006/05/05 09:46
10	BRS	L10	0	7 and align$ adj5 operands	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	2006/05/05 09:47
11	BRS	L11	0	7 and aligning adj5 operands	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	2006/05/05 09:47
12	BRS	L12	154	aligning adj5 operands	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	2006/05/05 09:48
13	BRS	L13	0	7 and 12	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	2006/05/05 09:47

14	BRS	L15	3	1 and 14	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	2006/05/05 09:49

	Type	L #	Hits	Search Text	DBs	Time Stamp
15	BRS	L14	35	aligning adj operands	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	2006/05/05 09:50